NASA Computational Case Study
Characterizing Moving Particles

Nargess Memarsadeghi and Brent J. Bos
NASA Goddard Space Flight Center, Greenbelt, MD, 20771.

Studying small particles is of interest to NASA scientists for many applications such as characterizing the lunar and Martian dust environments. In this homework, we characterize the motion of small particles via image analysis.

NASA, primarily the Goddard Space Flight Center, has developed a large depth-of-field Particle Image Velocimeter (PIV), with a collection of image analysis based algorithms for particle characterization. NASA developed it to characterize fluxes of dust particles in the Martian atmosphere in particular and planetary surfaces in general [3, 4]. The technology has many potential terrestrial applications, too, including monitoring of airborne industrial pollutants and airborne particles in mine shafts. For small particles, 150 μm in diameter

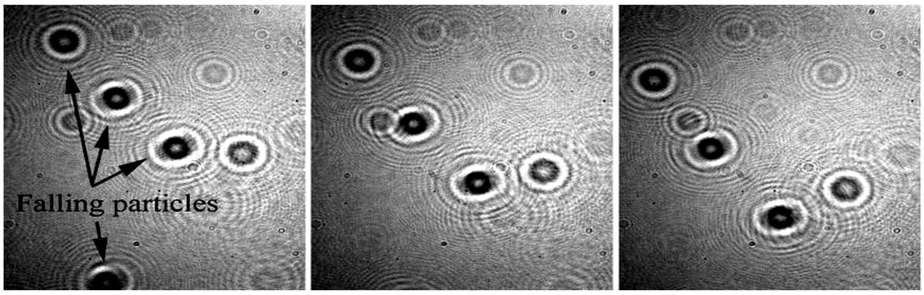

Figure 1: During a 2007 field campaign on the Santa Cruz Flats in Arizona, Dr. Brent Bos used his Large Depth-of-Field Particle Image Velocimeter (PIV) to capture these images of desert dust particles as small as 10 microns in diameter in free-fall.

or smaller, the dust signatures that PIV captures are dominated by rotationally symmetric near-field diffraction patterns, as shown by the falling particle patterns in Figure 1. Several image processing algorithms were designed and developed for PIV in order to characterize dust particles. Characterization algorithms include counting dust particles, determining their locations, sizes, as well as direction and velocity of their movement [3]. In this case study, we will learn how optical properties of traveling dust particles are used to estimate their velocity vectors via image analysis.

We first study the optical properties of stationary and moving dust particles. Then, we learn how to utilize these properties to determine the travel direction and velocity of a moving dust particle.

Stationary Dust and Motion Blur

Stationary dust particles typically appear as nested circular patterns in PIV images. We work with a simulated dust particle of size $250\mu m$, Image I, that is available on the website in the file `SIMPBYTSCL15.tiff` .

Activity 1. Read data values stored in the `SIMPBYTSCL15.tiff` file. Report the number of rows, columns, and type of the data. Update these data values by subtracting image's background value from all its pixels. Finally, place these values in the center of a two-dimensional 512×512 array of all zeros. We call this final product image I.
Hint 1: You can use the MATLAB routine called `imread`, or implement its equivalent, for reading .tiff files.
Hint 2: The background value is the value stored at position (1,1) of the original image.

In the next two activities, we simulate how a dust particle moving horizontally might appear in an image.

Activity 2. Construct Image B of the same size and data type as Image I, such that all its pixels are black except a horizontal line of seven pixels with magnitude $\frac{1}{7}$, located at the center of the image. We call B a *horizontal filter*. Convolve Image I with Image B. How does the resulting image, BI, compare to image B?
Hint: Use `conv2` routine of MATLAB or implement its equivalent.

Activity 3. Repeat the same exercise with horizontal filters of lengths equal to 5, 9, 11, and 13 pixels each. Save and display the results as .tiff images. How does the filter length affect the results?
Hint 1: What happens to the brightest and darkest areas of the image?
Hint 2: How does the shape of the dust particle relate to the filter size?

Now we investigate motion in other directions.

Activity 4. Consider again the horizontal filter of seven pixels centered at the middle of the image. Convolve the original image I with this filter rotated, such that it makes a 30, 60, and then 90 degree angle with the horizontal line passing through the center of the image. Display and save the results in .tiff images. How does the rotation affect the results?
Hint 1: Use the `imrotate` routine of MATLAB or implement its equivalent to get an image of the rotated filter with the same dimensions as the original image.
Hint 2: Where are the brightest areas in the results? In what orientation do these areas appear?
Hint 3: Does the particle elongate in any particular direction?

Moving Dust and Deblurring

We just blurred a clear image I with several blurring filters B. When an image is blurry, it means that the original clear image was convolved with some blur function or filter. If we have some knowledge of the properties of the blurring function or filter, we can restore the clear image. We refer to this process as *deblurring*. One can use the information about the blurring filter or function to *deconvolve* the blurred image and retrieve the original clear image. For moving particles, the blurring filter is directly related to the velocity vector of that particle. That is, the direction and velocity of moving dust particles contribute to the appearance and shape of their blurry images. Thus, if we estimate the blurring filter correctly, we then know the direction and velocity of the traveling particle.

Exercise. Suppose we know the blurring filter. Derive the mathematical equation yielding the clear image from the blurred image and its blurring filter. What is a potential problem with using this approach for estimating the clear image?
Hint: Use the Convolution theorem and properties of the Fourier transforms [5]. The convolution theorem states that the Fourier transform of the convolution of two matrices is the Schur (entry-wise) product of their Fourier transforms. In other words, if C is obtained by convolving matrix A with matrix B, then the following holds.

$$\begin{aligned} C &= A * B \\ \textsc{fft}(C) &= \textsc{fft}(A) \times \textsc{fft}(B), \end{aligned}$$

where * represents the convolution and $\times$ represents the Schur product. Use these equations to find A given B and C.

Activity 5. Consider the blurred image `bImage.tiff`. Display the blurred particle in gray scale values. Estimate the travel direction of the dust particle.
Hint 1: The travel direction of a dust particle is the same as the direction of its blurring filter.
Hint 2: Use what you learned from Activity 3 about the relationship between the direction of the filter and properties of the blurred image of a dust particle.

Activity 6. Deblur the blurred dust particle saved in `bImage.tiff`. Display and save the clear image in a .tiff file. Report the travel velocity of this particle.
Hint 1: The travel velocity is a function of the blurring filter's length. Report the velocity in terms of the blurring filter's length in pixels.
Hint 2: Use the Convolution theorem and your answer to the Exercise section.
Hint 3: Use your results from Activity 4 regarding the travel direction of the particle.
Hint 4: Once you know the direction of the filter, all you need to do is estimate its correct length. Assuming the exposure time of the camera can result in filters of size 1 to 15 pixels before a particle moves out of the frame completely, find the length of the blurring filter in pixels. That is, do an exhaustive search among possible filter lengths and choose the one that minimizes an appropriate error function.
Hint 5: Place filters at the center of an image of size 512×512 with all background pixel values equal to zero.

In this case study, we learned how to characterize the motion of dust particles. Our solutions were motivated by optical properties of stationary and moving dust particles. We derived velocity estimates using the mathematics of linear algebra and optimization, important tools for image analysis. Alternative algorithms make use of similar tools.

Challenge. Can you deblur the same image using another technique? Can you do so without making any assumptions about the direction or length of the filter? What assumptions did you make? How well does your algorithm work in presence of noise in the blurred image?
Hint: You may benefit from publications on Lucy-Richardson filter [2, 6], Wiener-Helstorm filter [8, 9], or deblurring algorithms [7].

References

[1] Matlab Central. `http://www.mathworks.com/matlabcentral/`. Accessed: July 22, 2011.

[2] D. S. C. Biggs. Acceleration of iterative image restoration algorithms. *Applied Optics*, 36(8):1766–1775., 1997.

[3] B. J. Bos, S. Antonille, and N. Memarsadeghi. Characterization of moving dust particles. In C. A. Bouman, I. Pollak, and P. J. Wolfe, editors, *IS&T/SPIE Electronic Imaging, Computational Imaging IX*, volume 7873, pages 78730E1–78730E9, San Francisco, CA, United States, January 2011.

[4] B. J. Bos and W. M. Farrell. Dust flux instrumentation for Mars landers. In *7th International Conference on Mars*, Pasadena, CA, July 2007.

[5] R. Bracewell. *The Fourier Transform and Its Applications.* McGraw-Hill, NY, third edition, June 1999.

[6] R. J. Hanisch, R.L. White, and R.L. Gilliland. *Deconvolution of Images and Spectra.*, chapter 10: Deconvolution of Hubble Space Telescope Images and Spectra., pages 310–356. Academic Press, Boston, MA, 2nd edition, 1997.

[7] P. C. Hansen, J. G. Nagy, and D. P. O'Leary. *Deblurring Images: Matrices, Spectra, and Filtering.*, volume 3 of *Fundamentals of Algorithms.* Society for Industrial and Applied Mathematic., 1st. edition, 2006.

[8] R. L. Easton Jr. *Fourier Methods in Imaging.* The Wiley-IS&T Series in Imaging Science and Technology. John Wiley and Sons, Ltd., 2010.

[9] J. R. Schott. *Remote Sensing: The Image Chain Approach.* Oxford University Press, 2nd. edition, 2007.